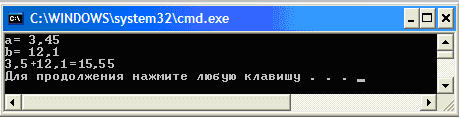
**ПРИЛОЖЕНИЕ А**

**Программирование алгоритмов линейной структуры на языке С#**

**Задание №1.** Написать программу, которая запрашивает с клавиатуры два вещественных числа, и выводит на экран сумму данных чисел (вещественные числа выводятся с точностью до 2 знаков после запятой):



Листинг программы:

static void Main(string[] args)

{

WriteLine("-------------------------------------");

WriteLine("Введите вещественное число А: ");

double a = ToDouble(ReadLine());

WriteLine("Введите вещественное число B: ");

double b = ToDouble(ReadLine());

WriteLine("-------------------------------------\n");

var sum\_of\_num = a + b;

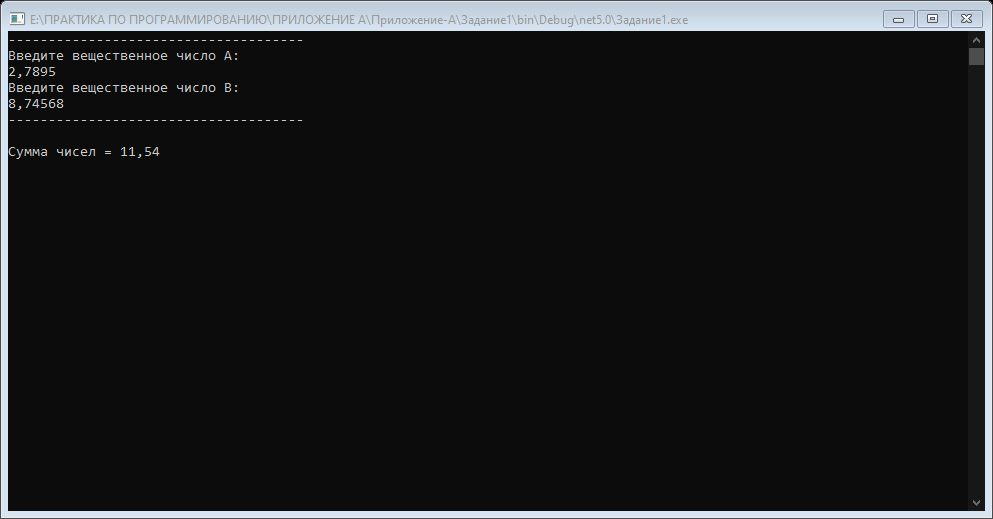
WriteLine($"Сумма чисел = {Math.Round(sum\_of\_num,2)}");

ReadKey();

}Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| A=2,7895  B = 8,74568 | 11,54 |

Анализ результатов:

Рисунок 1.1 – Результат работы программы

**Задание №2. Написать программу**: Дано трехзначное число, в котором все цифры различны. Получить шесть чисел, образованных при перестановке цифр заданного числа.

Листинг программы:

static void Main(string[] args)

{

try

{

WriteLine("Введите трёхзначное число: ");

var myStringNumber = ReadLine();

if (myStringNumber.Length < 3 || myStringNumber.Length > 3)

{

throw new Exception("Длина числа не соответствует условию! ");

}

else

{

var myIntNumber = ToInt32(myStringNumber);

int a, b, c;

List<int> digits = new List<int>();

static void GetDigit(int number, List<int> digits)

{

while (number != 0)

{

var d = number % 10;

digits.Add(d);

number /= 10;

}

}

GetDigit(myIntNumber, digits);

foreach (var digit in digits)

{

WriteLine($"Цифра: {digit}");

}

string[] massiv = new string[] {"123", "132", "213", "231", "312", "321" };

for (int elem = 0; elem < massiv.Length; elem++)

{

var mask = massiv[elem];

var i1 = (int)char.GetNumericValue(mask[0]);

var i2 = (int)char.GetNumericValue(mask[1]);

var i3 = (int)char.GetNumericValue(mask[2]);

var number = digits[i1 - 1] \* 100 +

digits[i2 - 1] \* 10 +

digits[i3 - 1];

WriteLine(number);

}

}

}

catch (Exception myNumber)

{

WriteLine($"Ошибка: {myNumber.Message}");

}

finally

{

ReadKey();

}

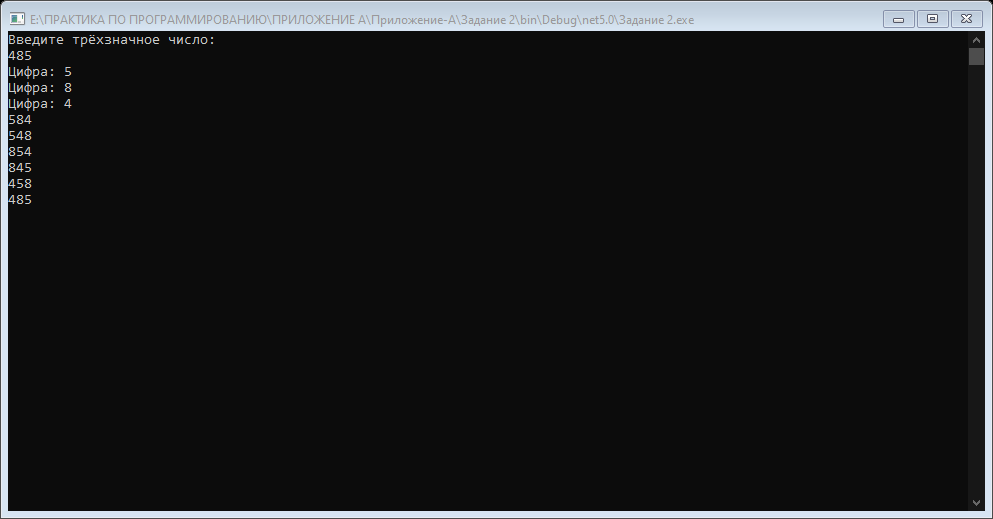
}

}

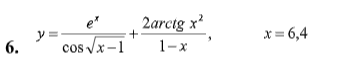
Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 485 | 584, 548, 854, 845, 458, 485 |

Анализ результатов:

Рисунок 1.2 – Результат работы программы

**Задание №3**. Вычислить значение функции для заданного значения аргумента:



Листинг программы:

static void Main(string[] args)

{

const double e = 2.7182818284;

const double x = 6.4;

try

{

var y = (Math.Pow(e, x) / Math.Cos(Math.Sqrt(x - 1))) + (2 \* Math.Atan(Math.Pow(x, 2)) / 1 - x);

WriteLine($"Ответ: {y}");

}

catch (Exception ex)

{

WriteLine(ex.Message);

}

ReadKey();

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| e = 2.7182818284  x = 6.4 | -883,4213336125849 |

Анализ результатов:

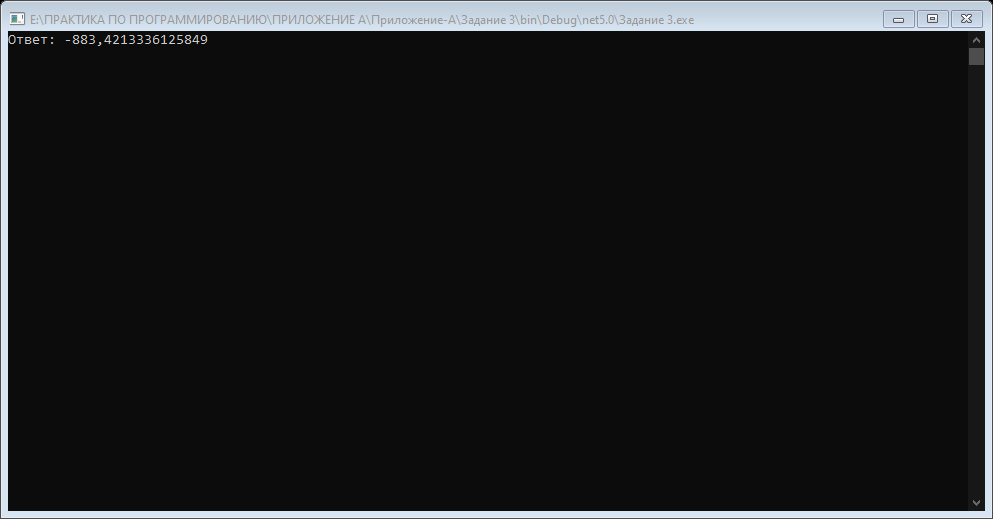


Рисунок 1.3 – Результат работы программы

**Задание №4.** Определить расстояние S и скорость v, пройденное физическим телом за время t, если тело движется с постоянным ускорением а и имеет в начальный момент времени скорость v0.

Листинг программы:

static void Main(string[] args)

{

try

{

WriteLine("Введите скорость в начальном моменте времени (v0): ");

double v0 = ToDouble(ReadLine());

WriteLine("Введите постоянное ускорение (a): ");

double a = ToDouble(ReadLine());

WriteLine("Введите время (t): ");

double t = ToDouble(ReadLine());

var V = v0 + a \* t;

var S = v0 \* t + (a \* Math.Pow(t, 2) / 2);

WriteLine($"Скорость = {V} м/c");

WriteLine($"Расстояние = {S} м");

}

catch (FormatException)

{

WriteLine("Неправильный ввод данных!");

}

catch (DivideByZeroException ex)

{

WriteLine(ex.Message);

}

finally

{

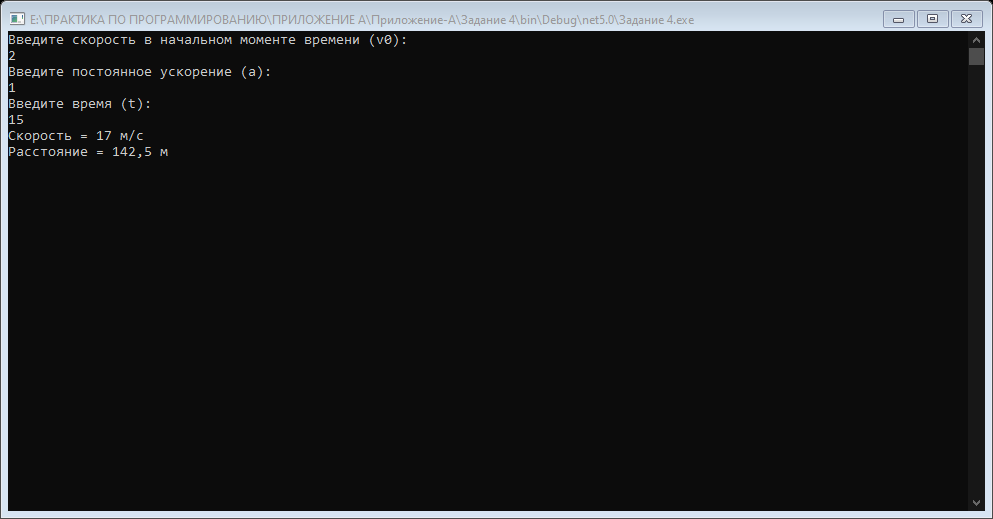
ReadKey();

}

Таблица 1.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| v0 = 2  a = 1  t = 15 | Скорость = 17 м/c  Расстояние = 142,5 м |

Анализ результатов:

Рисунок 1.4 – Результат работы программы